# An Architecture Supporting Safety Critical Software

Soon-Key, Jung

## Samenvatting

De keuringsinstanties zijn erg conservatief in het goedkeuren van veiligheidsgerelateerde systemen waarvan het gedrag uitsluitend door programmatuur gecontroleerd wordt. In het algemeen wordt geen veiligheidskeurmerk afgegeven voor extreem veiligheidskritische systemen gebaseerd op software met hoge complexiteitsgraad. Er bestaat een aantal methoden en richtlijnen die hun nut reeds hebben bewezen bij de ontwikkeling van programmatuur voor de besturing van veiligheidskritische technische processen. Voordat zulke software in gebruik wordt genomen, wordt ze uitgebreid getest ter verificatie en validatie. Bij de huidige stand van de ontwikkelingen zijn deze testprogramma's echter niet in staat om de formele correctheid van grotere programma's met mathematische precisie te garanderen.

Het doel van dit proefschrift is het geven van een oplossing voor deze ongewenste situatie. Daartoe wordt een speciaal, en noodzakelijkerwijs eenvoudig, computersysteem ontwikkeld, dat in staat is veiligheidsgerelateerde funkties uit te voeren binnen de context van gedistribueerde procesbesturingssystemen of programmable logic controllers (PLC's). Gebruikmakend van algemeen geaccepteerde methoden, wordt de hardware zodanig ontworpen, dat ze correct werkt en een 'fault tolerant' gedrag vertoont. De hardware kan bijvoorbeeld gebaseerd worden op de VIPER chip, waarvan de correctheid reeds formeel aangetoond is. De nadruk ligt hier echter op de programmatuur, omdat de betrouwbaarheid van de software nog steeds achterblijft bij die van de hardware. De hier gepresenteerde methode is uniek, omdat zij de eerste poging vormt om op het niveau van de architectuur ondersteuning te bieden voor de verificatie van de software.

De architectuur biedt volledige temporele voorspelbaarheid, determinisme en supervisie over de programma-executie en over alle andere activiteiten van het computersysteem. Zij biedt tevens expliciete ondersteuning van een specifieke software-verificatiemethode, nl. 'diversitaire terug-documentatie' en van sequentiële controle, omdat verscheidene procesbesturingsprogramma's, die veiligheidsgerelateerde taken bevatten, sequentieel van aard zijn. De firmware bevat een minimaal besturingsprogramma. Een onderzoek in de chemische industrie heeft uitge-

wezen dat alle procesbesturingsprogramma's opgebouwd kunnen worden uit een kleine verzameling (ongeveer 40) functionele modules ("software IC's"), waarvan de correctheid formeel aangetoond kan worden dankzij de geringe complexiteit. Dit resultaat moet echter nog in enkele andere industrieën geverifieerd worden. De toepassing-specifieke modules moeten worden geïdentificeerd, grondig geverifieerd en hun code moet worden opgeslagen in ROM's. Met zulke module-bibliotheken kunnen veiligheidsgerelateerde procesbesturingsprogramma's grafisch worden geconstrueerd door het verbinden van modules, met ondersteuning van een speciaal CAD-programma. Het is daarmee nog steeds niet mogelijk om de vertaler, die de grafische representatie omzet in de object-code, te verifiëren, maar dit is ook niet noodzakelijk, omdat voor de verificatie van een bepaald applicatieprogramma alleen de connecties tussen de modules geverifieerd dienen te worden. Voor deze taak kan de genoemde, door de architectuur ondersteunde, methode van 'diversitaire terug-documentatie' efficiënt en economisch worden ingezet.